Aggies off the Rails

Antonia Jones, Brian Ta, Guillermo Lopez,
Ivan Rupert, Jose Ramos & Joshua Kutsch

THE TEXAS A&M UNIVERSITY SYSTEM
1876

Iteration 3: Table of Contents

1 Team Information 1

1.1 Team Roles 1

1.2 Customer Meeting date/time/place for demo 1

2 User Stories implemented 2

3 Design Diagram 3

4 User story documentation / UI changes 4

5 Changes to User Stories 6

6 Evaluation of code and tests 7

7 Pivotal Tracker / Github Repository / Heroku 8

1 Team Information

1.1 Team Roles

Product Owner:
Scrum Master:

1.2 Customer Meeting date/time/place for demo

Meeting 4: ? ??? 2017 at 7pm in Dr. Huang's office, HRBB 416

2 User Stories implemented

We have implemented the following five user stories in iteration 2:

Login:
As a student
So that I can access my class assignments
I want to login using a tamu email and an instructor assigned password

cucumber test:

```
cucumber features/sign_in_student.feature
```

Upload a Howdy Registered Students File:
As a logged in instructor
So that the students will see assignments posted to their courses
I want to upload the file of registered students from Howdy to the website that will parse it and add the students to the appropriate courses.

cucumber test:

```
cucumber features/register_student.feature
```

Upload a submission:
As a logged in student
So that I can submit an assignment to the instructor and get a grade
I want to add submissions to the database

cucumber test:

```
cucumber features/submit_assignment.feature
```

View past submissions:
As a logged in student
So that I can know what I turned in and when
Display database data about what assignments this student has uploaded submissions for

cucumber test:

```
cucumber features/view_past_submissions.feature
```

Create an Instructor Account:
As an administrator
So that new instructors can use the website
Create a new instructor account using their email

cucumber test

```
cucumber features/create_instructor.feature
```

3 Design Diagram

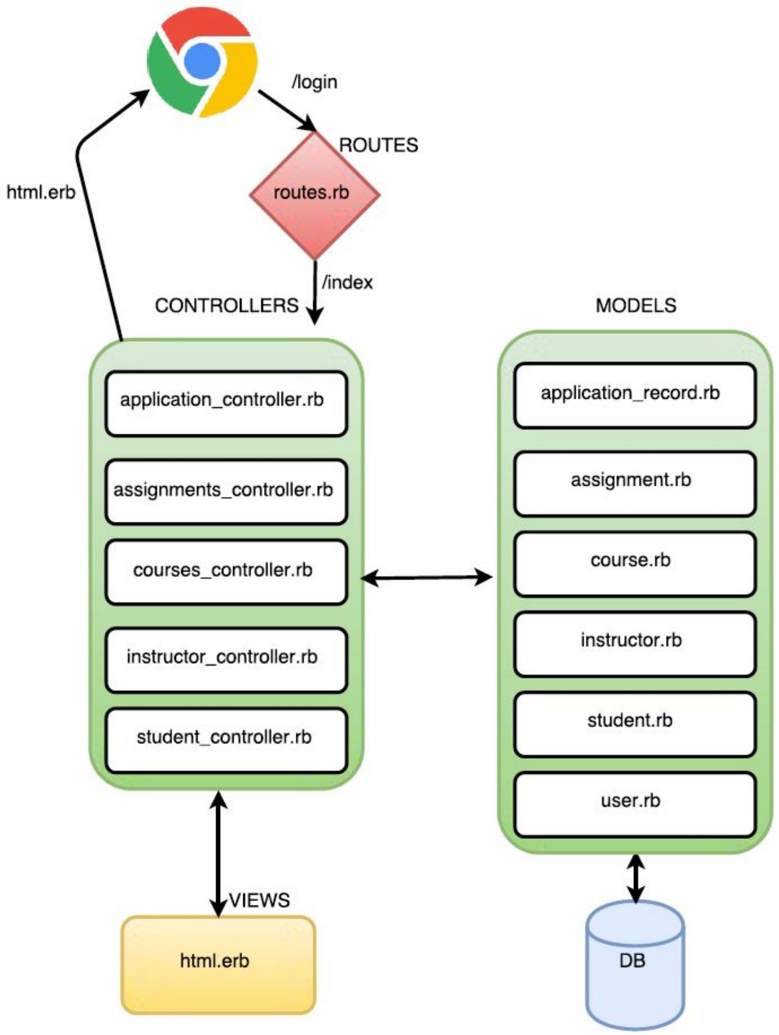


Figure 1: A high level design diagram of CSNET 2.0

4 User story documentation / UI changes

→ The five user stories for this iteration have been implemented completely. A user of this app can log in as a student, upload an assignment as a student, view past submissions as a student, and as a logged in instructor can add more students to your class.

→ Some changes were made to the lo-fi UIs, but they were not very significant. In our current iteration we believe that the user is given all of the relevant information they need for the features.

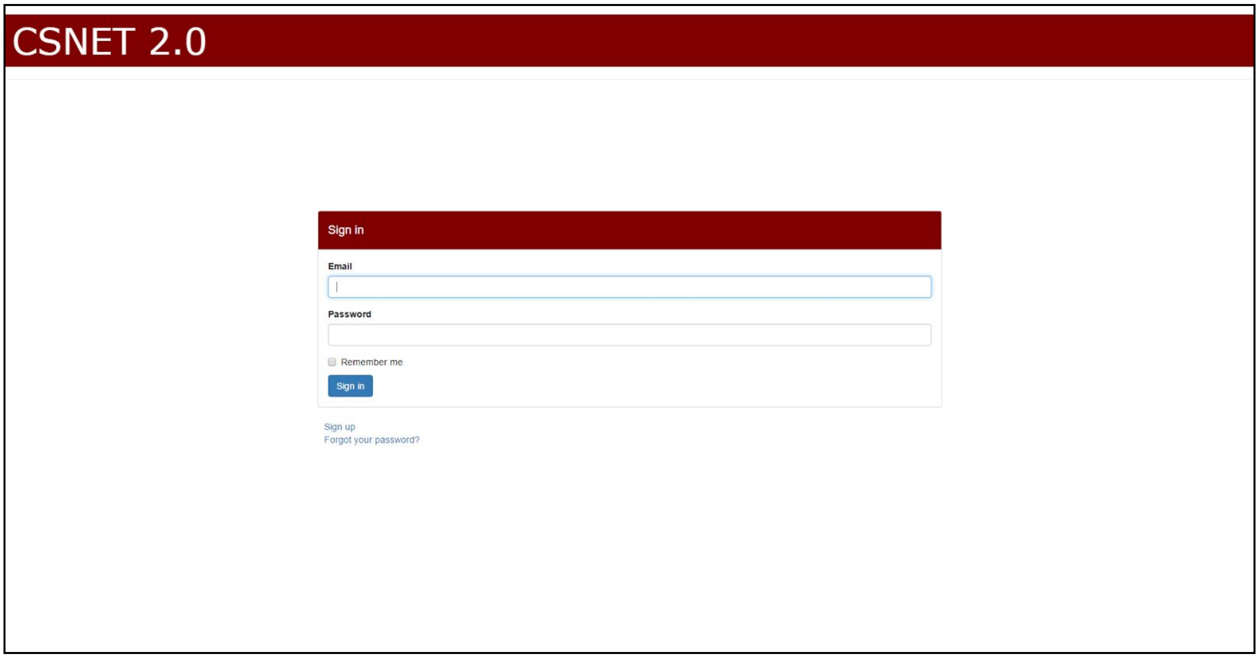
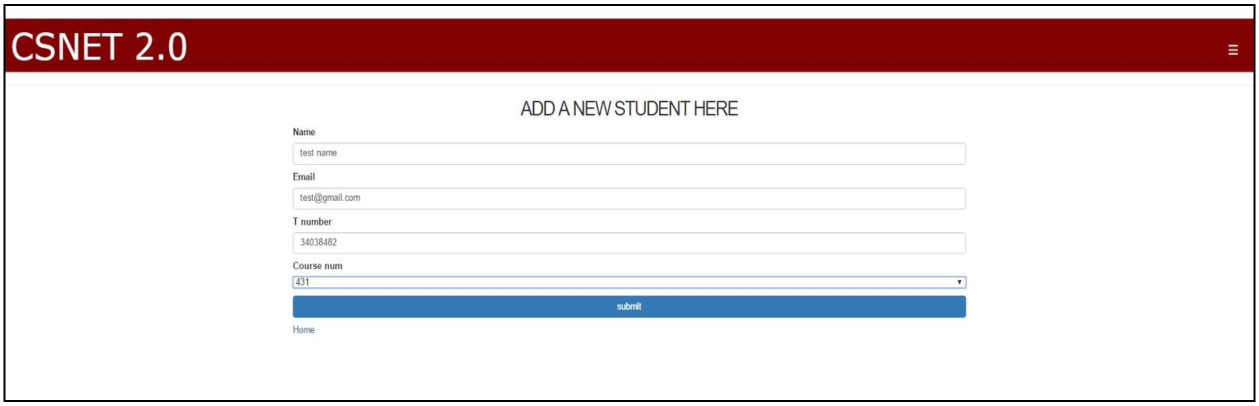


Figure 2: A screenshot of the student login page



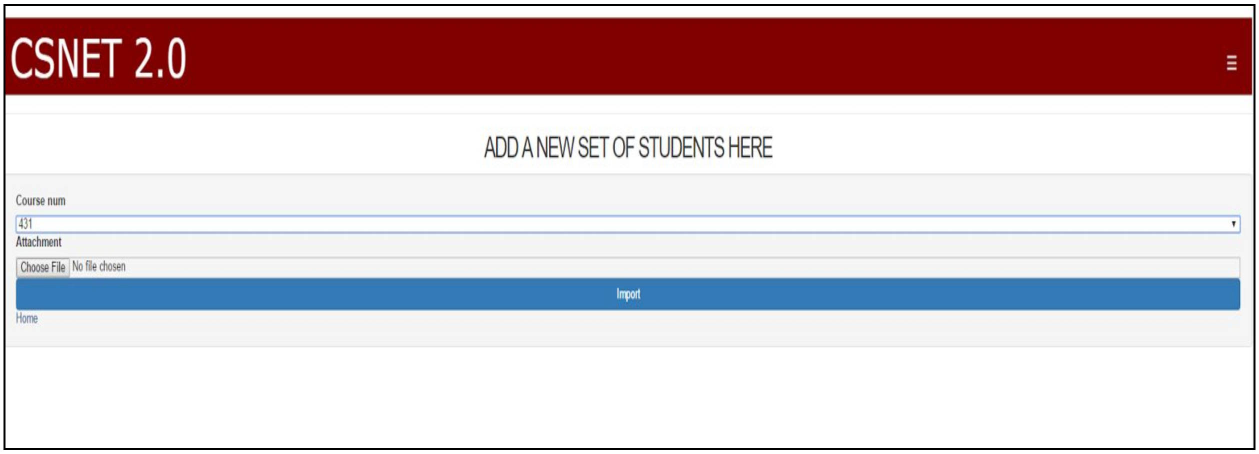
Name: test name

Email: test@gmail.com

T number: 34038482

Course num: 431

Figure 3: A screenshot of an instructor adding a student manually



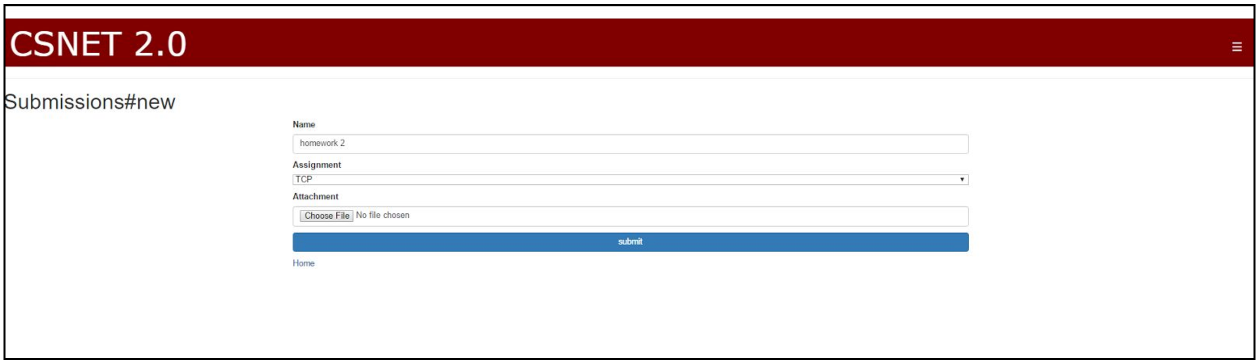
CSNET 2.0

ADD A NEW SET OF STUDENTS HERE

Course num: 431

Attachment: No file chosen

Figure 4: A screenshot of an instructor adding a set of students from an excel document.



CSNET 2.0

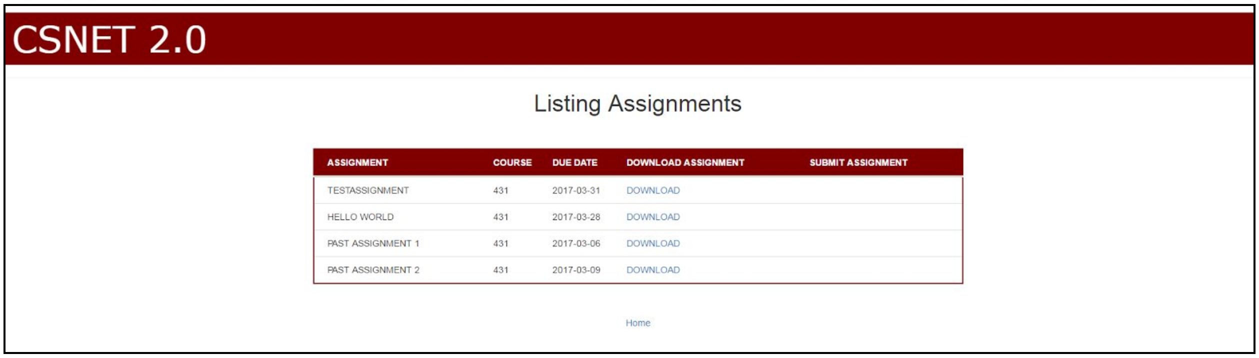
Submissions#new

Name: homework 2

Assignment: TCP

Attachment: No file chosen

Figure 5: A student submitting an assignment



CSNET 2.0

Listing Assignments

ASSIGNMENT	COURSE	DUE DATE	DOWNLOAD ASSIGNMENT	SUBMIT ASSIGNMENT
TESTASSIGNMENT	431	2017-03-31	DOWNLOAD	
HELLO WORLD	431	2017-03-28	DOWNLOAD	
PAST ASSIGNMENT 1	431	2017-03-06	DOWNLOAD	
PAST ASSIGNMENT 2	431	2017-03-09	DOWNLOAD	

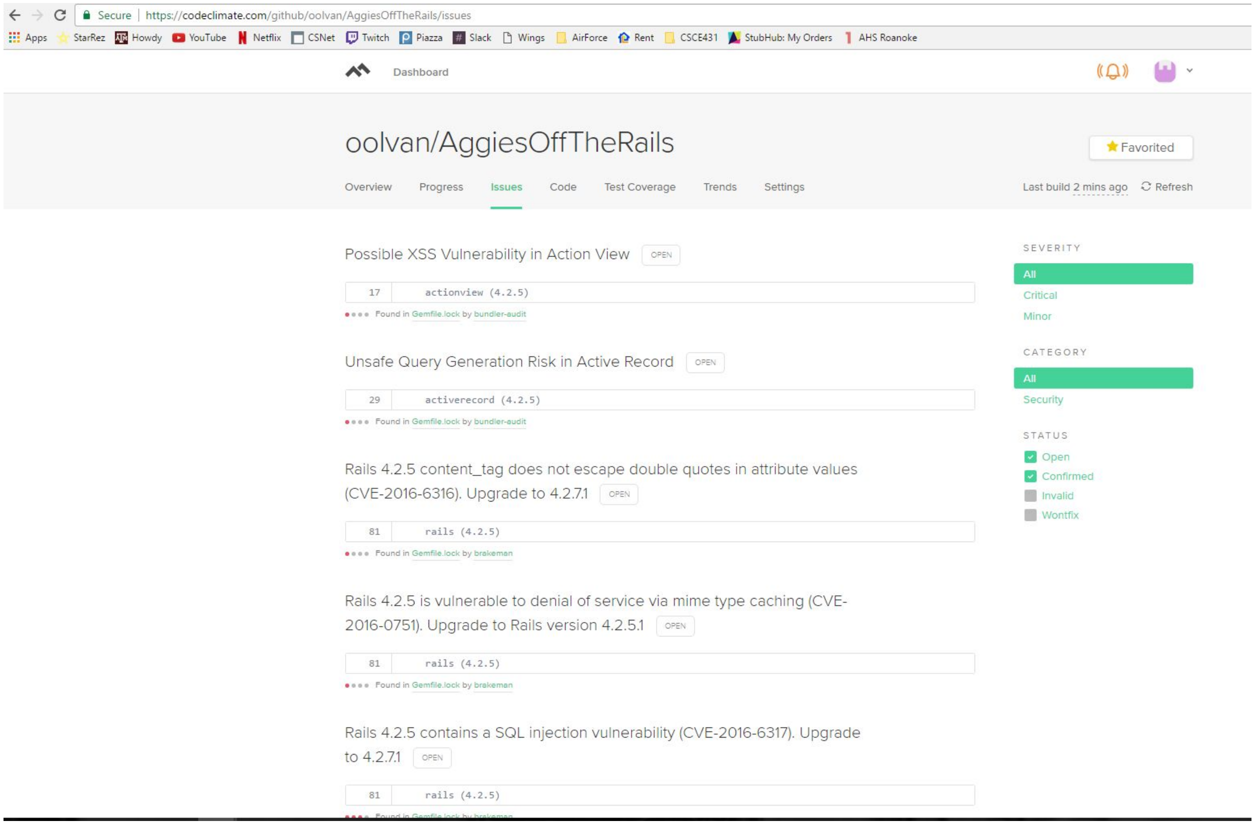
Figure 6: A student viewing his/her assignments

5 Changes to User Stories

→We have allowed for an instructor to upload an excel (.xls) file in order to add multiple students to a course at once. We have also added an administrator account that could add other instructors to the website.

6 Evaluation of code and tests

Below is an evaluation of our code from CodeClimate:



oolvan/AggiesOffTheRails

Possible XSS Vulnerability in Action View OPEN

```
17    actionview (4.2.5)
```

Found in Gemfile.lock by bundler-audit

Unsafe Query Generation Risk in Active Record OPEN

```
29    activerecord (4.2.5)
```

Found in Gemfile.lock by bundler-audit

Rails 4.2.5 content_tag does not escape double quotes in attribute values (CVE-2016-6316). Upgrade to 4.2.7.1 OPEN

```
81    rails (4.2.5)
```

Found in Gemfile.lock by brakeman

Rails 4.2.5 is vulnerable to denial of service via mime type caching (CVE-2016-0751). Upgrade to Rails version 4.2.5.1 OPEN

```
81    rails (4.2.5)
```

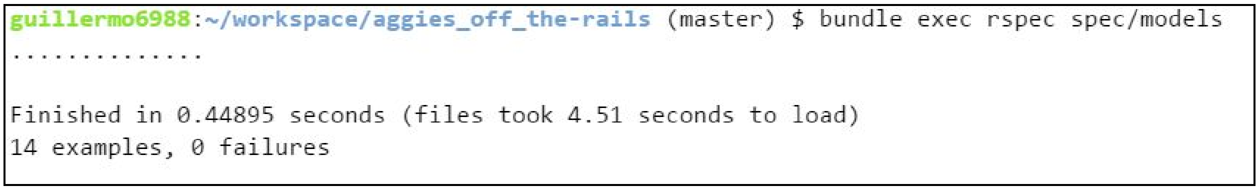
Found in Gemfile.lock by brakeman

Rails 4.2.5 contains a SQL injection vulnerability (CVE-2016-6317). Upgrade to 4.2.7.1 OPEN

```
81    rails (4.2.5)
```

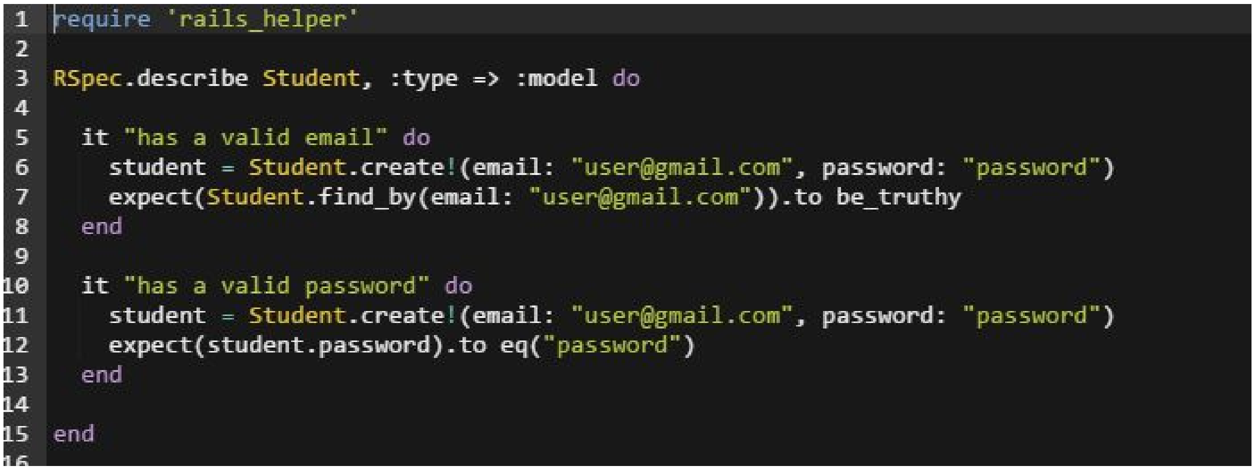
Found in Gemfile.lock by brakeman

Our rspec testing includes testing the Student, Roster, and Submission models as follows**:**



```
guillermo6988:~/workspace/aggies_off_the-rails (master) $ bundle exec rspec spec/models
..............

Finished in 0.44895 seconds (files took 4.51 seconds to load)
14 examples, 0 failures
```



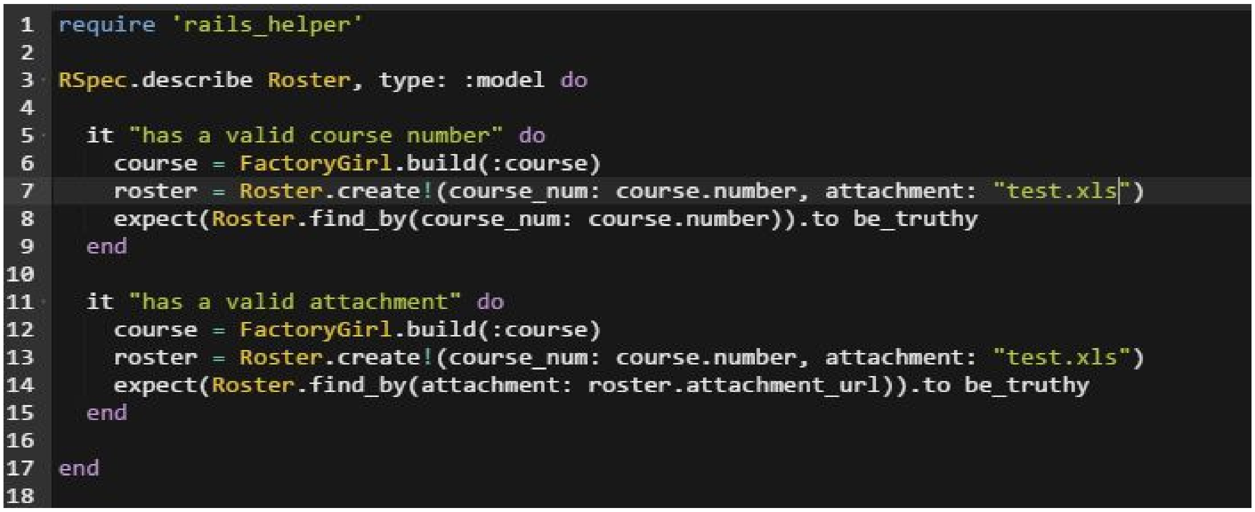
```ruby
require 'rails_helper'

RSpec.describe Student, :type => :model do

  it "has a valid email" do
    student = Student.create!(email: "user@gmail.com", password: "password")
    expect(Student.find_by(email: "user@gmail.com")).to be_truthy
  end

  it "has a valid password" do
    student = Student.create!(email: "user@gmail.com", password: "password")
    expect(student.password).to eq("password")
  end

end
```



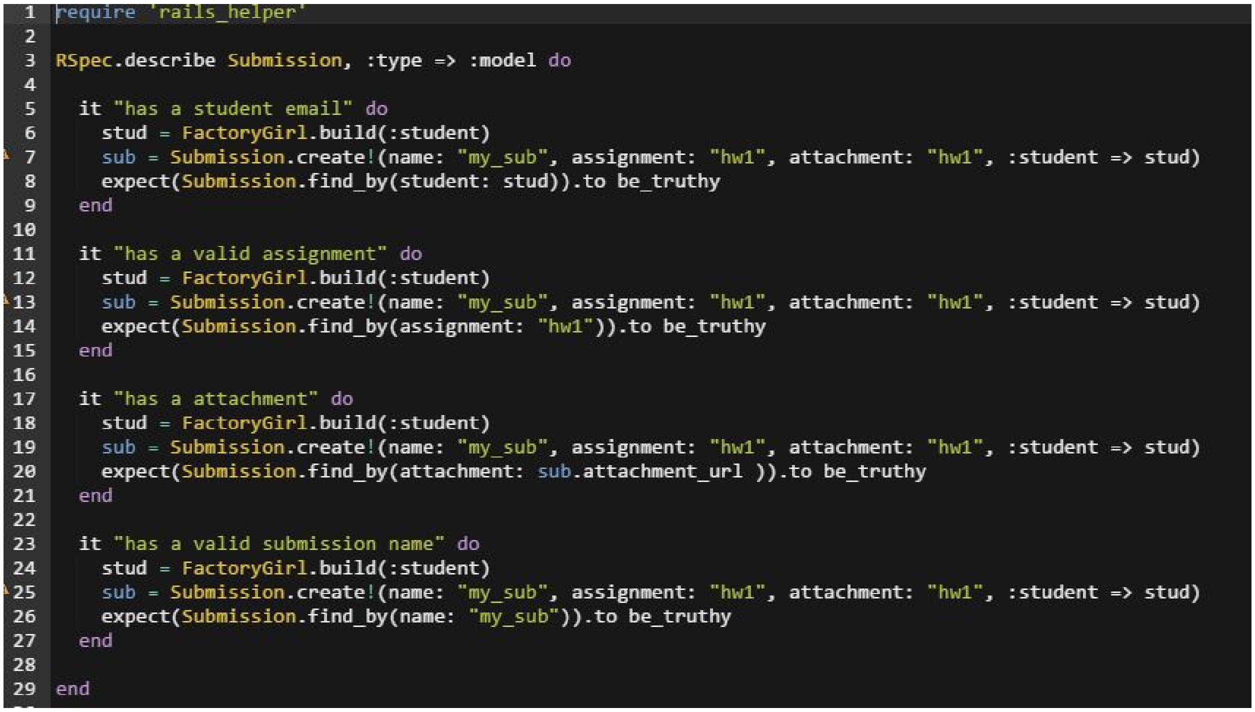
```ruby
require 'rails_helper'

RSpec.describe Roster, type: :model do

  it "has a valid course number" do
    course = FactoryGirl.build(:course)
    roster = Roster.create!(course_num: course.number, attachment: "test.xls")
    expect(Roster.find_by(course_num: course.number)).to be_truthy
  end

  it "has a valid attachment" do
    course = FactoryGirl.build(:course)
    roster = Roster.create!(course_num: course.number, attachment: "test.xls")
    expect(Roster.find_by(attachment: roster.attachment_url)).to be_truthy
  end

end
```



```
require 'rails_helper'

RSpec.describe Submission, :type => :model do

  it "has a student email" do
    stud = FactoryGirl.build(:student)
    sub = Submission.create!(name: "my_sub", assignment: "hw1", attachment: "hw1", :student => stud)
    expect(Submission.find_by(student: stud)).to be_truthy
  end

  it "has a valid assignment" do
    stud = FactoryGirl.build(:student)
    sub = Submission.create!(name: "my_sub", assignment: "hw1", attachment: "hw1", :student => stud)
    expect(Submission.find_by(assignment: "hw1")).to be_truthy
  end

  it "has a attachment" do
    stud = FactoryGirl.build(:student)
    sub = Submission.create!(name: "my_sub", assignment: "hw1", attachment: "hw1", :student => stud)
    expect(Submission.find_by(attachment: sub.attachment_url )).to be_truthy
  end

  it "has a valid submission name" do
    stud = FactoryGirl.build(:student)
    sub = Submission.create!(name: "my_sub", assignment: "hw1", attachment: "hw1", :student => stud)
    expect(Submission.find_by(name: "my_sub")).to be_truthy
  end

end
```

7 Pivotal Tracker / Github Repository / Heroku

- Pivotal Tracker (login needed)

- Aggies off the Rails GitHub Repository

- Aggies off the Rails Heroku App